Introduction to Swirles

and HPC in general

Juliana Kwan, 5 Nov 2025

Maths IT team

How to get help

- I am a Research Software Engineer attached to GR (Paul Shellard) and Astro. I mostly work on numerical relativity with some ML/AI applications.
- HPC support: Marek Szuba and Merlin Hartley
- IT manager: Matthijs van den Bergh
- Please do not email them directly, use help@maths.cam.ac.uk to open a support ticket.
- Office hours: ???

How to write a good support ticket

Get help faster

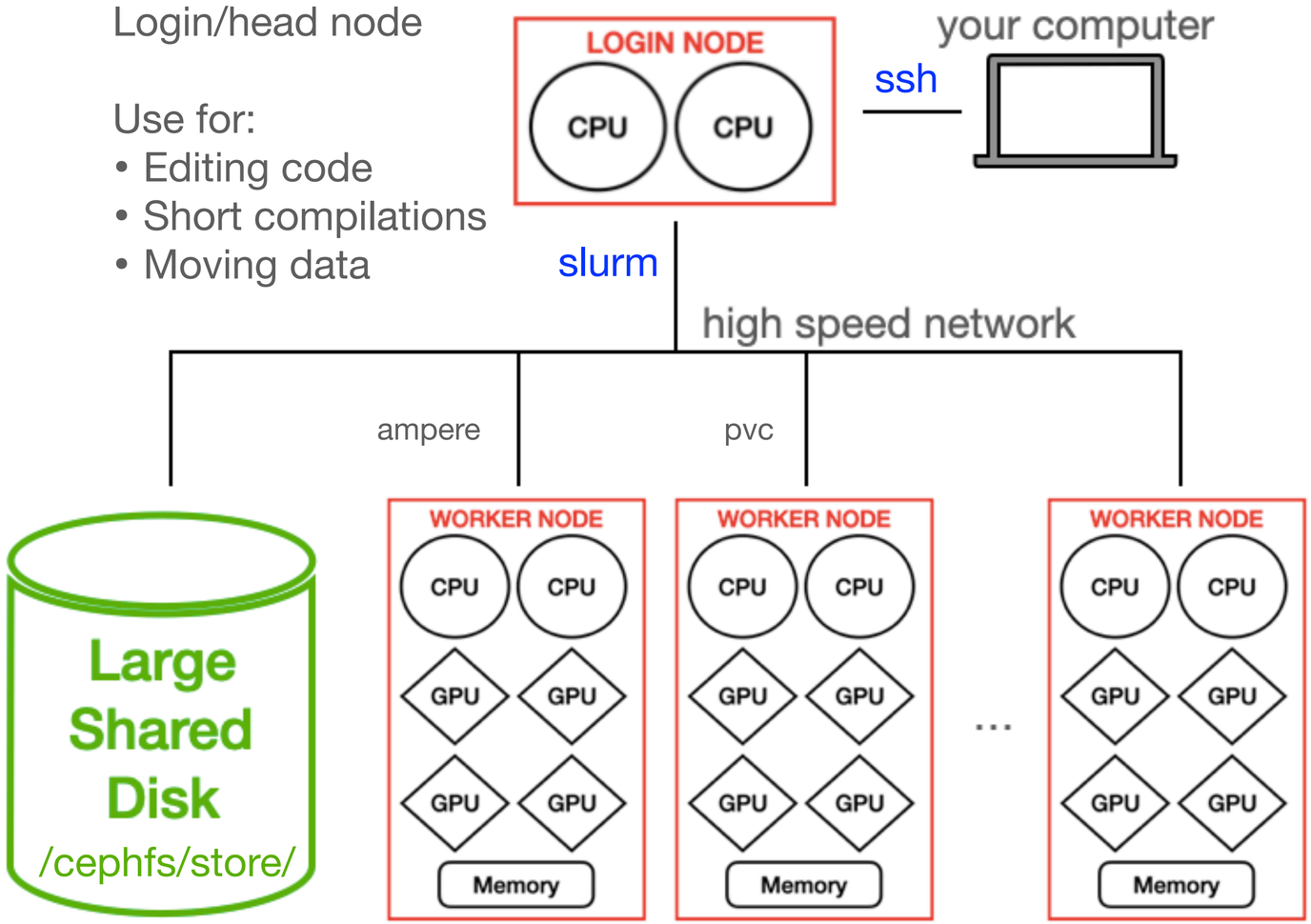
- Please include as many details as you can, such as
  - What is the machine doing? What should the machine be doing?
  - Any error messages (and lines preceding the error)
  - Details of job configuration, e.g. are you using MPI, OpenMP, any GPUs
  - Environment variables (use `env`)
- Normal rules of etiquette apply!

Intro to HPC systems

Some basic terminology

Login/head node

Use for:
- Editing code
- Short compilations
- Moving data

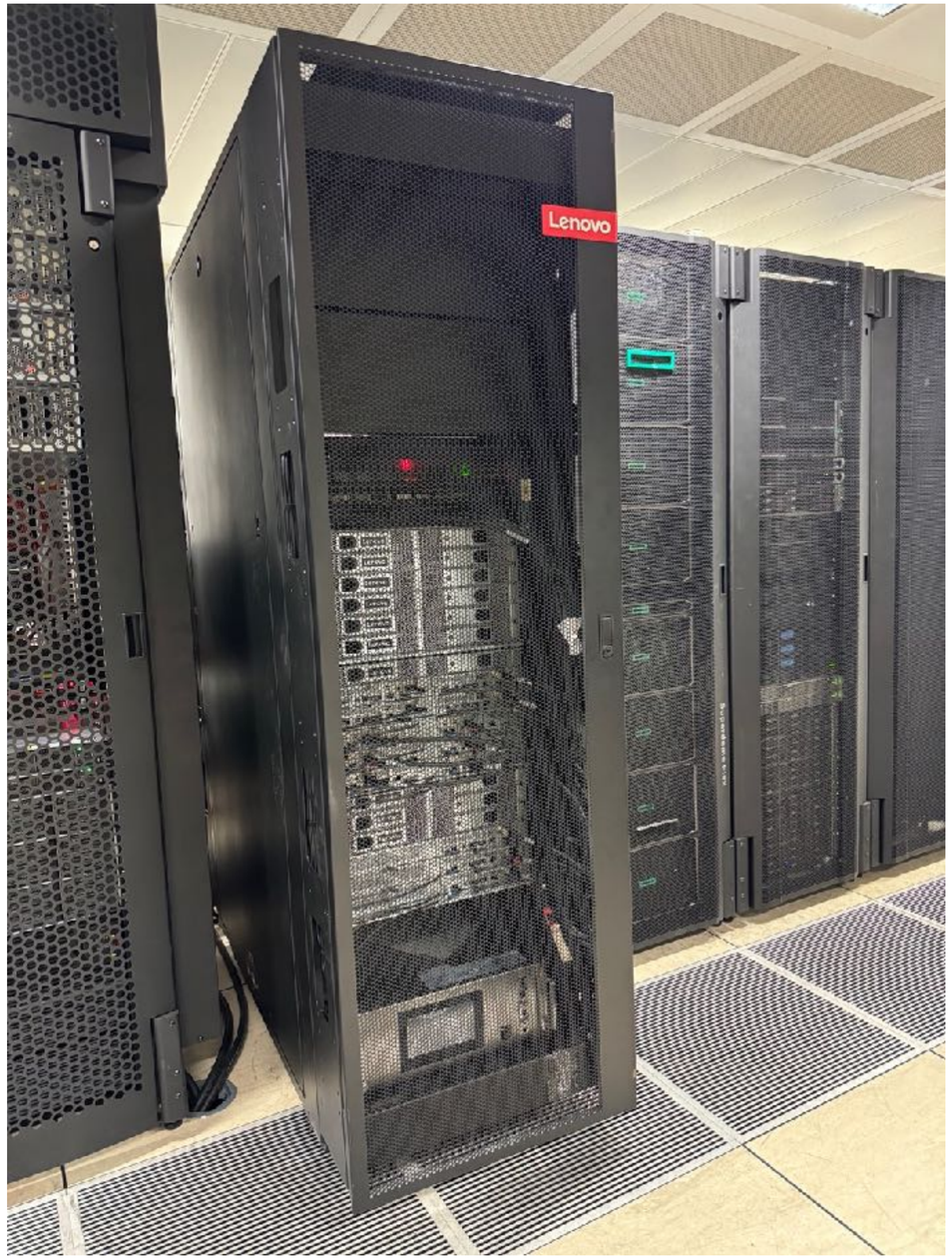


Compute nodes

Use for:
- All computational work
- Long compilations
- Compilations requiring a specific environment, e.g. -xHost

What is Swirles?

- Swirles is our new HPC cluster and the successor to Fawcett.
- This is the official documentation for Swirles
- There are several partitions (use `sinfo` to check):
  - The Intel components are named Cosmos XI
    - `pvc`: 3 nodes of Intel Data Center GPU Max 1550 (PVCs), Sapphire Rapids host with 1TB mem.
    - `spr`: 4 nodes of Sapphire Rapids HBMs, 1TB men
  - `ampere`: 3 nodes of Nvidia A100, host has 2TB mem
  - `lovelace`: 1 node of Nvidia L40
  - `genoa`: 6 nodes of AMD EPYC 9654, 1.5 TB mem
  - `cosmosx`: this used to be Fawcett, Intel Xeon Gold 6154, single node, 32 sockets, 18 cores/socket
- Intel parts funded by ExCALIBUR as a GPU testbed



Access to Swirles

- First generate a SSH key-pair using: `ssh-keygen -t ed25519 -f <name_of_ssh_key>`

replacing <name_of_ssh_key> with something memorable. It will prompt you for a password to access the key pair, please do not leave it blank. The key pair needs to go in your ~/.ssh directory and the public key needs to be sent to help@maths.cam.ac.uk

You will only be able to login once your access has been confirmed by the Maths IT helpdesk.

- How to log in:
  - In your .ssh/config file, define a new entry:

```
Host swirles.maths.cam.ac.uk
User <CRSid>
ProxyJump username@ssh.maths.cam.ac.uk
IdentityFile ~/.ssh/<name_of_ssh_key>
```

  Then you can directly log into Swirles using `ssh CRSid@swirles.maths.cam.ac.uk`

- Your username is your CRSid, the password is your usual Maths password for logging into your desktop or any Unix workstation. It is not necessarily the same as the one for your email or Raven.
- How to check if you have a login (and how long your account is valid for):

  https://useradmin.maths.cam.ac.uk/selfservice/record

- If you don't have a login, you need to contact Maths IT with a support request: help@maths.cam.ac.uk Please also mention which group you belong to so that your usage can be charged correctly and so that you can be assigned the appropriate storage space.

How to login
(hands on exercise)

1. Open a terminal
2. In your favorite text editor, open your ~/.ssh/config file
3. Add a new entry for Swirles:

```
Host swirles.maths.cam.ac.uk
User <CRSid>
ProxyJump username@ssh.maths.cam.ac.uk
IdentityFile ~/.ssh/<name_of_ssh_key>
```

5. Type `ssh <CRSid>@swirles.maths.cam.ac.uk`
6. Enter your Maths password
7. Enter your password for your Swirles ssh key

Slurm on Swirles

How to submit jobs

- Slurm is the software used to manage resources on Swirles, including allocating nodes to users.
- When you first login into Swirles, you will land on the head/login node. The head node is not to be used for computationally intensive tasks. Use the head node to submit jobs to the compute nodes via Slurm.
- Use Slurm to queue a job with sbatch or request an interactive node with srun/salloc

sbatch

Use this to submit a job script

Most hands-off option
Use if you have many jobs that are similar e.g. parameter file
Use if you need to set up a particular environment
Outputs into slurm-XXXX.out

```
sbatch my_script.sh
```

srun

Use this for debugging

Jobs will run on a compute node, but outputs to your terminal (on the head node)

Can be interactive if used with --pty bash flag

```
srun -p spr --nodes=1 —cpus-per-task=28 --time=00:30:00 <my_prog>
```

salloc

Use this for an interactive job

When resources have been allocated, you can ssh directly into compute node

```
salloc -p spr --nodes=1 --cpus-per-task=28 -—time=00:30:00
```

Example Slurm batch script

- Get this from GitHub:

git clone https://github.com/COSMOS-CTC-Cambridge/swirles-training.git

- Things to note:
  - This is a normal bash script. Usual bash commands like `source` will work.
  - Slurm will inherit the same environment from the session in which the script was submitted
  - Slurm settings are prefaced with `#SBATCH`
  - At a minimum, you need to set the partition and the number of nodes and the time
  - srun is used to call the executable. No other flags necessary as job details will be inherited from `#SBATCH`

```
#!/bin/bash

#SBATCH --partition <name>         #<name> can be either pvc, ampere, genoa, spr
#SBATCH --nodes=<#nodes>           #how many nodes
#SBATCH --ntasks=<#tasks>          #how many tasks (MPI ranks) in total
#SBATCH --cpus-per-task=<#cpus>    #set to > 1 to use OpenMP threads
#SBATCH --time=<hh:mm:ss>
#SBATCH --gres=gpu:<#gpus>         #necessary for GPU resources, choose from 1 - 4

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

cd ${MY_WORK_DIR}
source my_modules.sh

srun --mpi=<..> ./my_exec
```

Note that the --mpi flag needs to be set to pmi2 for Intel MPI and pmix for OpenMPI.

Slurm on Swirles

Some advanced options

- Slurm has its own set of environment variables prefaced by `SLURM`, e.g.
  - You can use `SLURM_CPUS_PER_TASK` to set the number of OpenMP threads
- To use Intel MPI:
  - You need to set `--mpi=pmi2` to launch Intel MPI jobs with the proper communication protocol.
  - The PVCs (Intel GPUs) will only use Intel MPI
- To use OpenMPI:
  - You need to set `--mpi=pmix` to launch OpenMPI jobs
  - Use OpenMPI with Nvidia GPUs (Recommended)
- Inside a batch script, you can launch jobs with `srun` or `mpiexec/mpirun` or `torchrun`
  - Usually `srun` is best
  - Can use `mpiexec/mpirun` with Intel MPI (for extra options for exposing PVC tiles etc.)
  - Always use `torchrun` with `PyTorch` (it will figure out the optimal MPI settings for you)

Slurm on Swirles
(hands on exercise)

- Clone the GitHub repo: git clone https://github.com/COSMOS-CTC-Cambridge/swirles-training.git
- Go to the Examples directory
- Edit the batch script:
  - Ask for 1 node and 4 tasks on Sapphire Rapids partition for 10 mins
- Submit the job script:
  - `sbatch example_script.sh`
- Check on your job using `squeue -u <CRSid>`
- Check on the queue using `sinfo -p <partition>`
- Look at the output in slurm-XXXX.out

The Swirles environment

How to gain access to pre-installed software

- Swirles uses a module system, like on Fawcett.
- Some useful module commands:
  - `module list` - list currently loaded modules
  - `module avail` - list available modules
  - `module load <software>` - load <software> into your $PATH
  - `module display <software>` - shows information on <software>, install location, other environment variables set
  - `module use <path>` - access modules installed in <path>, module avail will show these modules
  - `module save <name>` - save your currently loaded set of modules under <name>
  - `module restore <name>` - restore the modules saved above
  - `module savelist` - list the names of saved module environments
  - `module purge` - remove all loaded modules.
- No modules are loaded by default (try doing a module list)
- Some essential modules:
    - `module load cmake`
    - `module load miniforge3/4.8.3-4-Linux-x86_64/gcc/s2ikv54u`
    - `module load python` **(NB: this automatically loads a number of other Python packages, which may be incompatible with your needs).**
      - Then use a virtual env to install the python packages that you need.

Compilers on Swirles

- For an Intel oneAPI environment:
  - `module load intel/compiler-intel-llvm/2025.1.1`
  - `module load intel/mpi/2021.15`
  - `module load intel/mkl/2025.1`
  - The MPI compiler is then mpiicc/mpiicpx/mpiifx for C/C++/Fortran
- For a GNU environment:
  - `module load gcc-runtime/11.4.0/gcc/3i5fbkgx`
  - `module load openmpi/5.0.3/gcc/xs76solb`
  - The MPI compiler is then mpicc/mpicxx/mpif90 for C/C++/Fortran
- For CUDA:
  - `module load cuda/12.6.3/gcc/bxp4amlg`
  - The compiler is `nvcc`
  - Note that there is a problem using CUDA with MPI.
- NB: You can type ‘`which mpirun`’ to check your version of MPI

Spack on Swirles
(for advanced users)

- Spack is a package manager (like conda, virtual environments) but is also a way to bundle software (like pip).
- You do not need admin privileges to install packages with Spack
- Clone the spack repository: `git clone --depth=2 https://github.com/spack/spack.git`
- Source the spack environment: `. spack/share/spack/setup-env.sh`
- New packages can be installed with `spack install <package>`
  - If you need a particular version of e.g. hdf5: `spack install hdf5@1.14`
- Packages can be loaded with `spack load <package>`
- There is extensive documentation: https://spack.readthedocs.io/en/latest/index.html

Storage on Swirles

Where do I put my stuff?

- Your home directory is `/cephfs/home/<CRSid>` - NB: this is not the same as Fawcett.
- You should have space for outputs on `/cephfs/store/<group-name>/<CRSid>` - each group has 20TB.
- To check your quota use:

```
getfattr -n ceph.quota.max_bytes /cephfs/home/<CRSid>
```

Compile and submit a job

(hands on exercise)